Running Head: LDAP

LDAP: Strengths, Weaknesses, and AD Integration

Jered McClure

Walden UniversityLDAP: Strengths, Weaknesses, and AD Integration

Lightweight Directory Access Protocol (LDAP) is “a set of open protocols used to access centrally stored information over a network,” and “runs on a layer above the TCP/IP stack” (Red Hat, Inc., 2011) (Mcirosoft Corporation, 2011). The use of directory structures in computer networking was a natural extension of the telephone directory structure which already existed when networking first began. LDAP’s strengths are many, and are what keep it in service to this day. However, it does have its weaknesses, as well.

LDAP entries are based on a hierarchical structure where one type can be used to describe many lower types (e.g. Company as a top type can be used to describe Google and Microsoft, which can then be broken down into further department types, business unit types, team types, and eventually employee types) (Microsoft Corporation, 2011). Furthermore, each entry has horizontal attribute descriptions which define the identification/function of the entity (e.g. fax number, phone number, name, location) (Red Hat, Inc., 2011).

Due to the flexibility of the API, and the ability of LDAP to support multiple database architectures, there are a range of options available on the market for LDAP deployment (Microsoft Corporation, 2011) (Red Hat, Inc., 2011). For instance, “Active Directory servers that support client applications using the LDAP API include Windows Server 2008, Windows Server 2003, and Windows 2000 server” (Microsoft Corporation, 2011). “The main benefit of using LDAP is that information for an entire organization can be consolidated into a central repository” (Red Hat, Inc., 2011).

“The associated LDAP API simplifies writing Internet directory service applications” (Microsoft Corporation, 2011). Because LDAP is an extensible protocol (think XML), changes can be made to it as needed by the organization utilizing its functions (Zeilenga, 2006). This goes back to the horizontal attributes of LDAP, allowing organizations to further expand on pre-existing identifiers as the need may arise.

Nevertheless, “LDAP cannot create directories or specify how a directory service operates” (Microsoft Corporation, 2011). This inability of LDAP is why there are such directory programs as Microsoft Active Directory (AD). “Microsoft provides the Active Directory Service Interfaces (ADSI) for developing client-side directory service applications” (Microsoft Corporation, 2011). “ADSI uses an LDAP provider to communicate with Active Directory” (Microsoft Corporation, 2011). While AD is the visual GUI component and directory creation method, it can be controlled and operated using LDAP protocols (hence why AD is setup in a hierarchical method).

All that being said, for every up there is a down. “When an LDAP client application connects to an LDAP server, it can either query a directory or attempt to modify it” (Red Hat, Inc., 2011). “If the client application is attempting to modify information within an LDAP directory, the server verifies that the user has permission to make the change and then adds or updates the information” (Red Hat, Inc., 2011). This means, at a key junction within the connection layer, security is left in the hands of the user/client.

The greatest threat to an LDAP system comes from spoofing and/or hijacking of an LDAP session. In spoofing, the user or client is tricked “into believing that the information came from the directory when in fact it did not, either by modifying data in transit or misdirecting the client’s transport connection” (R. Harrison, 2006). In hijacking, “An attacker seizes control of an established protocol session” (R. Harrison, 2006). Why these two are so dangerous is that they both rely on the user/client to identify that an attack is occurring. When the user is not aware of the signs involved with an attack (false site details, no HTTPS flag, and/or missing security information) is when attacks are successful.

System administrators overcome this by forcing user restrictions on what they can and cannot access, so that even if there is a security leak, the damage is minimal. Furthermore, proper network security in terms of firewalls, encryption, and threat management gateways ensure that these assaults remain in the “too hard” pile for most attackers. Even with these threats present, administrators still use the directory setup due to its organizational capabilities. LDAP offers too much in terms of system control and manageability to disregard.

As previously stated, LDAP can be implemented by a network administrator through Microsoft’s AD. This is done by setting up a domain which users/clients of a network resource connect. The Domain Controller (DC) identifies the user/client and authenticates their credentials. If successful, they are allowed access to the LDAP resources which exist in the AD environment. Going back to Proper network security, all connections to and from the DC must be encrypted and passed through a firewall. This threat management ensures the packets within the AD request are sound, and do not contain any dangers to the LDAP data. Thus, the network administrator ensures that the directory is secure.

Primarily, LDAP is an outline. A generality which companies like Microsoft can use to develop front end applications like AD to manage directory information. Without LDAP, they would have to come up with their own way of connecting their systems together, ways which may not be compatible with TCIP/IP, and therefore, not compatible with the internet as it is seen today. While the directory structure does have some flaws, they are greatly overshadowed by the strengths and organization which directory configuration supplies.

Reference

Mcirosoft Corporation. (2011, September 7). *Lightweight Directory Access Protocol*. Retrieved February 24, 2012, from MSDN: <http://msdn.microsoft.com/en-us/library/aa367008(VS.85).aspx>

Microsoft Corporation. (2011, September 7). *Directory Entries*. Retrieved February 24, 2012, from MSDN: <http://msdn.microsoft.com/en-us/library/aa366100(v=vs.85).aspx>

Microsoft Corporation. (2011, September 7). *LDAP and ADSI*. Retrieved February 24, 2012, from MSDN: <http://msdn.microsoft.com/en-us/library/aa366149(v=vs.85).aspx>

Microsoft Corporation. (2011, September 7). *Lightweight Directory Access Protocol*. Retrieved February 24, 2012, from MSDN: <http://msdn.microsoft.com/en-us/library/aa367008(VS.85).aspx>

Microsoft Corporation. (2011, September 7). *What is LDAP*. Retrieved February 24, 2012, from MSDN: <http://msdn.microsoft.com/en-us/library/aa367036(v=vs.85).aspx>

Microsoft Corporation. (2012). *Windows Management Instrumentation.* Retrieved February 17, 2012, from MSDN Library: <http://msdn.microsoft.com/en-us/library/aa394582%28VS.85%29.aspx>

R. Harrison, E. (2006, June). *RFC4513 - Lightweight Directory Access Protocol (LDAP): Authenticaiton Methods and Security Mechanisms*. Retrieved February 24, 2012, from RFC Editor: <http://www.rfc-editor.org/rfc/rfc4513.txt>

Red Hat, Inc. (2011). *13.2. LDAP Terminology*. Retrieved February 24, 2012, from Reference Guide for Red Hat Enterpirse Linux 4: Edition 4: <http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/4/html/Reference_Guide/s1-ldap-terminology.html>

Red Hat, Inc. (2011). *Chapter 13. Lightweight Directory Access Protocol (LDAP)*. Retrieved February 24, 2012, from Reference Guide for Red Hat Enterprise Linux 4: Edition 4: <http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/4/html/Reference_Guide/ch-ldap.html>

Zeilenga, K. (2006). *RFC4510 - Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map*. Retrieved February 24, 2012, from IETF Tools: <http://tools.ietf.org/html/rfc4510>